Module 7 | Ret2libc & ROP

Introduction

This group of videos closes the course giving you some insight into more advanced exploitation techniques.

Cybersecurity is an arms race. Malware authors are constantly trying to find new ways to exploit programs, while software developers and defensive cybersecurity experts are trying to devise ways to harden programs to make them more difficult to exploit. In this module, we'll close out the course by showing you two more modern exploit techniques, taking you through examples of how they're implemented, and describing why they were developed in the first place.

Learning Objectives

The objectives of this module, like most of the modules in this course, are to teach you how to design and execute malware as well as defend systems against it. We'll continue our focus on binary system exploitation in this module by looking at more advanced techniques used to exploit vulnerable programs.
The other objective is to learn how to compromise a program via programming flaws, just as in the previous module.
You'll use GCC, GDB, and python in this assignment. They should already have been installed in your Ubuntu LTS virtual machine. In this assignment you'll spawn a remote shell on your system via a flawed program that you'll write.

Required Instructional Materials

- Follow the video lectures
- Take notes on the provided slides
- Follow the material on your own system so you understand exactly what's going on

Summary

The 1990s called and it wants stack overflows back! Let's look at some more advanced techniques. Ret2libc is a technique developed after stack smashing that does not execute from the stack. Return oriented programming is a relatively new technique popularized in 2010 that uses an executables own code to build an exploit.

If you have questions about some aspect of Learn, **UNM LEARN Support** is available to troubleshoot technical problems.

Contact them 24/7 at 505-277-0857, 1-877-688-8817 or use the "Create a Support Ticket" link on the left Course Menu.